# MySQL数据库存储说明

1. MySQL数据存储结构

CREATE TABLE collect1(

id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

workcity VARCHAR(50) NOT NULL,

job\_name VARCHAR(100) NOT NULL,

job\_inwhichcompany VARCHAR(100) NOT NULL,

min\_salary VARCHAR(20) NOT NULL,

max\_salary VARCHAR(20) NOT NULL,

job\_category VARCHAR(20) NOT NULL,

workplace VARCHAR(100) NOT NULL,

zhaopin\_numbers VARCHAR(20) NOT NULL,

job\_welfare VARCHAR(100) NOT NULL,

education\_background VARCHAR(20) NOT NULL,

min\_workexperience VARCHAR(20) NOT NULL,

job\_form VARCHAR(20) NOT NULL,

job\_releasetime VARCHAR(20) NOT NULL,

job\_require TEXT NOT NULL,

company\_name VARCHAR(100) NOT NULL,

company\_form VARCHAR(20) NOT NULL,

company\_industry VARCHAR(100) NOT NULL,

company\_scale VARCHAR(30) NOT NULL,

company\_web TEXT NOT NULL,

company\_introduce TEXT NOT NULL,

company\_address VARCHAR(100) NOT NULL,

data\_addtime VARCHAR(20) NOT NULL,

data\_sourceweb TEXT NOT NULL,

PRIMARY KEY(id)

)ENGINE=InnoDB DEFAULT CHARSET=utf8

;

其中包括

**企业信息**：企业名称、企业形式、企业所属行业、企业规模、企业介绍、企业页面地址、数据添加时间、数据来源网站

**职位信息**：职位名称、职位工作城市、职位所属企业名称、工作性质 、最低月薪 、最高月薪、工作地点、职位发布时间、学历、招聘人数、最低工作经验、职位类型、职位要求、数据添加时间、数据来源网站

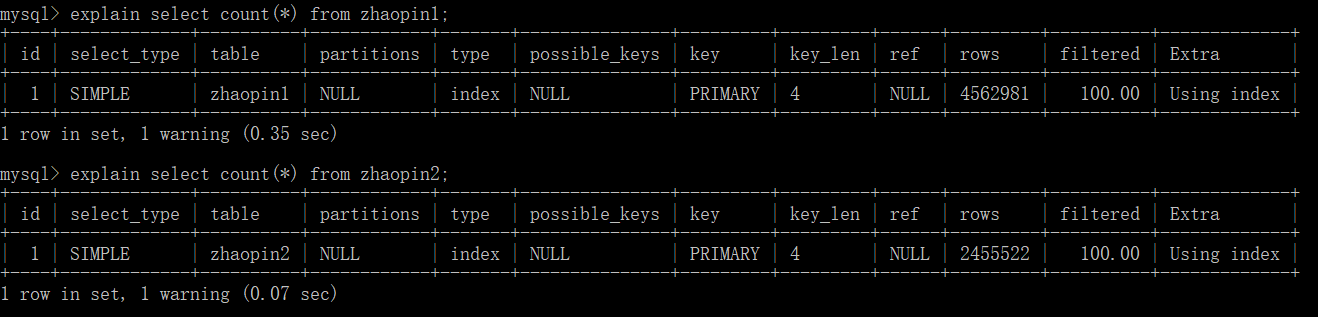
1. 数据库优化

大数据的存储使得数据的查询更新等操作变得很缓慢，于是进行了索引优化和分表优化。

1.分表优化：把数据分为多个表，并对这些表创建一个索引表，每个表的大小为500000行，对于5000000行的数据，即可拆分为10个表。使得查询等操作更快。

2.索引优化：对每一个表进行b-tree索引优化，其操作速度大幅度加快。对于500000行数据，即使采用二叉树构建，速度也可加快100倍以上。（实际建立的树取决于字段的类型大小，字段类型占的空间越小，树的高度越低，查询次数越少。）

1. 数据集数量

爬取了近4500000+2500000条数据存储于MySQL数据库中，

**数据集数据达700万条职位信息。**